Random Walks on Local Optima Networks

Marco Tomassini
Faculty of Business and Economics, Departement of Information Systems
University of Lausanne
Lausanne, Switzerland
marco.tomassini@unil.ch

***Abstract*—The Local Optima Networks represent combinatorial landscapes as graphs, where nodes are local optima and edges are transitions between optima. It brings a new set of metrics to characterize them. Here we investigate the behavior of random walks on such oriented and weighted networks using NK landscapes and QAP instances as examples. We show that random walks are useful to characterize the structure of the corresponding LONs and give interesting information about the relationships between search difficulty and LON structure.**

I. Introduction

Metaheuristic search methods (e.g, see [1]) make use of the concept of a fitness landscape [2] to find the global, or at least a good enough optimum, of difficult problems. The search takes place by going from a configuration to another or from a group of configurations to another through the use of suitable move operators. The full problem instance configuration space contains all the admissible solutions of a given problem. One can abstract from this space just the locally optimal configurations and the transitions between them. This view leads to the idea of Local Optima Networks (LONs) [3], [4] in which the nodes are the local optima of the underlying optimization problem and the edges account for the transitions among them using a neighborhood operator. The construction of the complete LONs requires the full enumeration of the optima of a given problem instance thus apparently limiting the approach to relatively small problem instances. However, sampling methods have been proposed that extend the method to larger spaces without loosing too much information [5]. LONs can also deal with neutrality in a search space, an important feature common to many combinatorial optimization problems [6], [7]. Modeling combinatorial landscapes as LONs brings a whole new set of useful metrics coming from complex network science [8] to capture the topology and structure of combinatorial search spaces and provides tools for estimating search difficulty.

In this contribution we present an application of random walks to LONs in order to characterize their structure. The chosen test combinatorial problems are NK landscapes and the Quadratic Assignment Problem (QAP). Random walks have been indirectly used in the study of LONs in [9] through the PageRank algorithm in the context of problem difficulty in NK landscapes. Random can be very useful to sample and explore properties of networks and in many other fields. Indeed, they have been used early to study population features starting from a single member and iteratively tracking a random neighbor at each time step [10]. Random walks are also at the basis of some network centrality measures such as random walk betweenness and PageRank [8], [11] and have been used in community detection too [12]. In the realm of fitness landscapes, fitness autocorrelation measures are based on random walks [13] and they are also used in large LONs sampling [5]. In this work, the application of random walks to LONs serves to characterize a number of hardness-dependent features of the corresponding problems, notably the frequency and mean waiting times with which high-fitness solutions are found in a way that has not appeared before.

The article is structured as follows. The next section briefly overviews the idea of local optima networks. Next, Sect. III reviews some fundamental notions on random walks that will be used in the sequel. Section IV describes the combinatorial landscapes considered for the study. Section V presents the results of our investigation and, finally, Section VI summarizes our findings and suggest directions for future work.

II. The local optima network model

The local optima network model for combinatorial landscapes was first proposed in [3]. For the sake of self-consistency, we give a summary of fitness landscapes and of local optima networks notions. A *fitness landscape* [14] is a triplet $(\mathcal{S}, \mathcal{N}, f)$ where $\mathcal{S}$ is a set of potential solutions *i.e.* a search space; $\mathcal{N} : \mathcal{S} \longrightarrow 2^{\mathcal{S}}$, a neighborhood structure, is a function that assigns to every $x \in \mathcal{S}$ a set of neighbors $\mathcal{N}(x)$, and $f : \mathcal{S} \longrightarrow \mathbb{R}$ is a fitness, also called objective, function that provides the objective value of the corresponding solutions.

For a given problem instance, the corresponding LON is obtained by extracting all the local optima in the instance's fitness landscape. These nodes form the set V of the vertices of the LON. The edges E then correspond to transitions between two given optima, weighted by their frequency. In more detail, the vertices V are exhaustively extracted using a best-improvement hill-climber (hc), as given in Algorithm 1 in the case of objective function maximization. Thus, a solution $x \in \mathcal{S}$ is a local optimum iff $\forall x' \in \mathcal{N}(x) : f(x') \leqslant f(x)$. For a minimization problem, the inequality is reversed. For some problems with large amounts of neutrality it may happen that some optima are actually plateaus, i.e., more than one solution has the same locally optimal objective function value. LONs generalize to this case [6] but we will not need this extension here as all the optima are unique configurations in our case.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

Algorithm 1 Best-improvement hill-climbing (maximization)

```
procedure HILLCLIMBING
    x ← random initial solution
    while x ≠ Local Optimum do
        set x' ∈ N(x), s.t. f(x') = max_{y∈N(x)} f(y)
        if f(x) < f(x') then
            x ← x'
        end if
    end while
end procedure
```

The edges E of the network, called *escape edges* [15], are defined according to a distance function $dist$ and a positive integer $D > 0$. The distance function represents the minimal number of moves between two solutions by a given search (mutation) operator. There is an edge e_{ij} between optima v_i and v_j if a solution x exists such that $dist(x, v_i) \leqslant D$ and $hc(x) = v_j$. In other words, if v_j can be reached after mutating v_i and running hill-climbing from the mutated solution. The weight $\tilde{w}_{ij}$ of this edge is $\tilde{w}_{ij} = |\{x \in \mathcal{S} \mid dist(x, v_i) \leqslant D \text{ and } hc(x) = v_j\}|$. That is, the number of v_i mutations that reach v_j after hill-climbing. This weight can be normalized by the total number of solutions, $|\{x \in \mathcal{S} \mid dist(x, v_i) \leqslant D\}|$, within reach at distance D: $w_{ij} = \tilde{w}_{ij} / \sum_j \tilde{w}_{ij}$.

LONs can be built exhaustively as above only for relatively small problem instances. As the number of optima and their connections grow quickly with problem size, the complete enumeration of optima becomes impractical, or even infeasible, for large enough instances. In this case, sampling must be used. In the present work we shall deal with problem sizes that allow full enumeration.

Summarizing, the weighted local optima network $G_w = (V, E)$ corresponding to a given problem instance landscape is the graph where the nodes $v_i \in V$ are the local optima, and there is an edge $e_{ij} \in E$, with weight w_{ij}, between two nodes v_i and v_j if $w_{ij} > 0$. The weight w_{ij} may be different than w_{ji} and G_w is a weighted and directed graph.

III. Random walks on weighted directed graphs

LONs are weighted and directed networks and thus we must use the appropriate methodology to model random walks on them. Here we only provide enough information so as to make the presentation self-contained; an extensive recent treatment of random walks on networks is found in [16]. In ordinary random walks on graphs the walker repeatedly moves from a given vertex to a neighboring vertex with a given probability. Consider the simplest case: an undirected and simple graph $G(V, E)$, i.e. without self-loops and multiple edges. In this case, when the walker is on a given node $k \in V$ and the set of neighbors of k is $\mathcal{N}_k$, the probability p_{kj} that the next visited node is $j \in \mathcal{N}_k$ is given by

$$p_{kj} = 1/|\mathcal{N}_k|$$

where $|\mathcal{N}_k| = d_k$ is the degree of vertex k. Clearly, $\sum_{j \in \mathcal{N}_k} p_{kj} = 1$.

The matrix $\mathbf{T}$ whose row elements p_{kj} are the above probabilities for each vertex k of the graph G is called the transition, or stochastic, matrix for the random walk process. An important question about these random walks is: starting with a given probability distribution $\mathbf{p}_0 = (p_1, \ldots, p_N)_0$ on the set V of N graph vertices, what will be the probability distribution $\mathbf{p}_n$ after n steps, or when $n \to \infty$? By definition, after one step the probability distribution will be $\mathbf{p}_1 = \mathbf{p}_0\ \mathbf{T}$ and from step j to $j + 1$ it changes as

$$\mathbf{p}_{j+1} = \mathbf{p}_j\ \mathbf{T} \tag{1}$$

Therefore, by iterating the previous equation from $j = 0$ to $j = n - 1$, the probability distribution $\mathbf{p}_n$ after n steps is:

$$\mathbf{p}_n = \mathbf{p}_0\ \mathbf{T}^n$$

that is, it is given by the initial probability distribution vector times the n-th power of the transition matrix $\mathbf{T}$. In the long time limit, if some conditions are satisfied, the probability distribution may reach an invariant value given by $\mathbf{p}^* = (p_1^*, \ldots, p_N^*)_\infty$ with $p_i^* = lim_{n\to\infty} p_i(n)$. Substituting in eq. 1 gives

$$\mathbf{p}^* = \mathbf{p}^* \mathbf{T} \tag{2}$$

From this eigenvalue equation the equilibrium, or stationary, probability distribution is the left eigenvector of $\mathbf{T}$ with eigenvalue 1 [16]. On a connected unweighted and undirected network it turns out that the probability p_k that a random walk will be found at node k at equilibrium is proportional to the degree d_k of node k, i.e., $p_k = d_k/2m$, where m is the number of edges. This results simply say that nodes of high degree are more likely to be visited in a random walk.

The application of random walks to LONs, where a node (optimum) may have self-loops as well as directed incoming and outgoing edges, requires some modifications. Since edges $\{kj\}$ are weighted (we assume $w_{kj} > 0$) and have a direction, transition probabilities p_{kj} are computed as follows:

$$p_{kj} = \frac{w_{kj}}{\sum_{i \in \mathcal{N}_k} w_{ki}}$$

in which the sum at the denominator is taken over all the *outgoing* links from k to its neighbors, a quantity that is called the *outgoing strength* of node k; k itself is included in the computation since in LONs each vertex may have a self-loop. It is easy to see that, for each node k, the rows of the matrix $\mathbf{T}$ defined above sum to 1, and thus $\mathbf{T}$ is a stochastic matrix. The simple weighted directed graph of Fig. 1 and the corresponding transition matrix $\mathbf{T}$ are meant to illustrate the ideas.

$$\mathbf{T} = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} 0.5 & 0.2 & 0.3 \\ 0.4 & 0.2 & 0.4 \\ 0.3 & 0.5 & 0.2 \end{pmatrix} \end{array}$$

The stationary distribution of an aperiodic and irreducible Markov chain as the random walks defined above can be found by solving equation 2 with an appropriate linear algebra algorithm. However, for large matrices there might be numerical

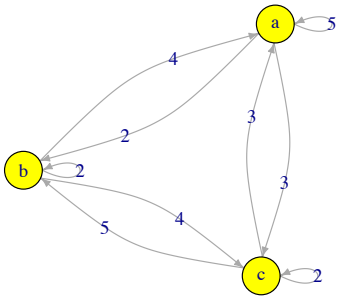

Fig. 1. A simple directed and weighted network.

difficulties that require specialized knowledge. Instead, here we shall follow a Monte Carlo simulation approach which, while it is much slower, is very simple and gives the same results in the long time limit. It consists in performing a large number of random walk steps starting from each network node and recording each time a given node is encountered. If the number of random walk steps is large enough, the final frequency with which each node has been hit will provide a very good approximation to the equilibrium occupancy probabilities.

Another quantity that can be computed from the random walks and that we will find useful is the *first hitting time*. For a given node v, the first hitting time t_v is the time step number n at which v has been first encountered in a random walk. Let us call $(v_1, v_2, \ldots, v_n, \ldots)$ the sequence of LON vertices encountered during a random walk. Then $t_v = min\{n \geq 0 : v_n = v\}$. The *mean first hitting time* is simply the above quantity averaged over all the random walks performed. The quantities described above depend on the LON structure and are expected to provide useful information about the problem and instance difficulty independent of any particular solution method.

IV. Problem description

In order to demonstrate the methodologies proposed in this study, we consider two well known families of combinatorial landscapes: the NK model [17] and the Quadratic Assignment Problem (QAP) [18]. We consider two standard solution representations: binary strings for NK landscapes and permutations for QAP. The single bit-flip operation changes a single bit in a given binary string in the NK case, whereas the pairwise exchange operation exchanges any two positions in a permutation, thus transforming it into another permutation for QAP. While NK landscapes are useful as a benchmark because they can be easily tuned from easy to hard, QAP is more representative of actual difficult combinatorial problems, even though artificial problems are used here as is the case in much of the literature.

A. The NK model

In a NK landscape [17] the model is a real random function Φ defined on binary strings $x \in \{0,1\}^N$ of length N, $\Phi : \{0,1\}^N \rightarrow \mathbb{R}^+$. The value of K determines how many other variables in the string influence a given x_i, $i = 1, \ldots, N$. The value of Φ is the average of the contributions ϕ_i of all the variable positions that are involved in the interaction :

$$\Phi(x) = \frac{1}{N} \sum_{i=1}^{N} \phi_i(x_i, x_{i_1}, \ldots, x_{i_K}),$$

and we search for the maximun value of Φ.

By increasing the value of K from 0 to $N-1$, NK landscapes can be tuned from smooth to rugged. For $K = 0$ all contributions can be optimized independently which makes Φ a simple additive function with a single maximum. At the other extreme, when $K = N-1$, the landscape becomes completely random. In this case, the probability of any given configuration being locally optimal is $1/(N+1)$, and the expected number of local optima is $2^N/(N+1)$. Intermediate values of K interpolate between these two extremes and have a varying degree of variable interaction [17].

The K variables that form the context of the fitness contribution of gene x_i are usually chosen according to the *random neighborhood* model, where the K variables are chosen uniformly at random among the $N-1$ variables other than x_i. In the following, instances of size $N = 14$ will be used because of the computational burden involved.

B. The Quadratic Assignment Problem

The Quadratic Assignment Problem (QAP) is a combinatorial problem in which a set of facilities with given flows has to be assigned to a set of locations with given distances in such a way that the sum of the product of flows and distances is minimized. A solution to the QAP is generally written as a permutation π of the set $\{1, 2, ..., n\}$. The cost associated with a permutation π is given by:

$$C(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{\pi_i \pi_j}$$

where n denotes the number of facilities/locations and $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ are referred to as the distance and flow matrices, respectively. The structure of these two matrices characterizes the class of instances of the QAP problem.

The results presented in this article are based on two instance generators proposed in [19] which are in turn inspired by [20]. The instance size of all instances used is 11, a relatively small value but one for which LONs can be computed exhaustively.

- *Uniform generator*: produces uniformly random instances where all flows and distances are integers sampled from uniform distributions. The distances are random integer numbers between 0 and 99. The flow matrix is symmetric with random integer entries between 1 and 99. This leads to the kind of problem known in literature as *Tai*nn*a*, nn being the problem dimension [20]
- *Real-like generator*: makes instances where the distance and flow matrices have structured entries. To generate the symmetric distance matrix, N points (integer coordinates)

are randomly distributed in a circle of radius 100, and the entries are given by the distances between these N points. The flow matrix is also symmetric with entries following the law $\lceil 10^r \rceil$ where r is a uniform random integer from $[L, U]$. This procedure generates non-uniformly random instances of type *Tai*nn*b* which have the so called "real-like" structure since they resemble the structure of QAP problems found in practical applications.

V. Random walks on LONs

In this section we present the results of running random walks on the LONs of instances of the two problems studied, i.e., the NK landscapes and the QAP.

A. Random walks on NK landscapes

Fig. 2 shows the equilibrium frequency $\mathbf{p}^*$ (see Sect. III) for each optimum in the LON of an NK instance with $N = 14$ and $K = 12$. The optima are ordered with fitness increasing to the right on the x-axis. The values are computed as the average occupation frequency after having executed $|V|$ random walks starting from each of the nodes of the LON each of length 1000 steps for a total of $10^3 \times |V|$ random walk steps, where $|V|$ is the number of nodes of the LON .

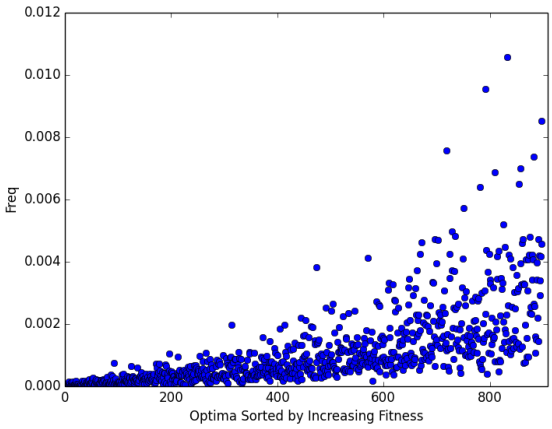

Fig. 2. Asymptotic frequency of visit of the LON nodes of an NK instance with $N = 14$ and $K = 12$ during a random walk. Node fitness increases from left to right on the x-axis.

It is clear that better optima on the right part of the image get visited much more frequently than less good ones on the left. Actually, the best solutions should be more difficult to find with a search heuristic in such a highly multimodal fitness landscape but it has been found in previous work that there is a positive correlation between fitness of a local optimum and the size of its attraction basin [4], [6] for all N and K values, two examples of which are given in Fig. 3 for an NK instance with $N = 18$ and $K = 4, 8$. This means that better solutions have larger basins which, translated into LON representation, implies that better optima have more incoming links from neighboring optima and this gives the random walk more chances to pass through the given maximum. Obviously, random walks are not recommended as a search technique

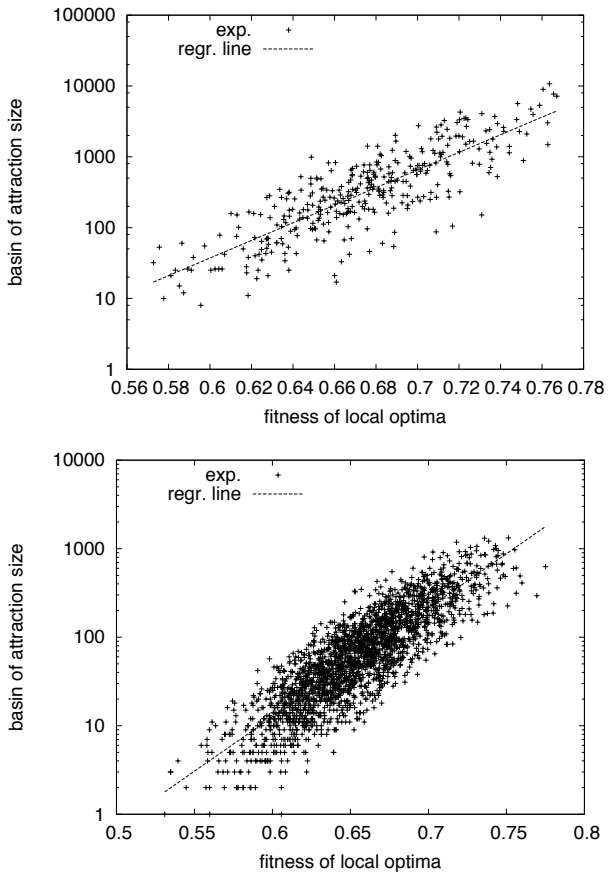

Fig. 3. Scatterplots of the fitness of local optima (x-axis) and their basin sizes (y-axis) for a representative instance with $N = 18$ and $K = 4$ (top image) and $N = 18$ and $K = 8$ (bottom image), with regression lines [4].

since the work to be done to populate all the optima increases exponentially with the problem size N and, for a given N, with increasing K, making it hopelessly slow for large N. However, the numerical computations leading to Fig. 2 show that for a quasi random landscape, i.e., with $K = N - 2$, the probability of finding an optimum by chance during a random walk is much higher for good optima than for bad ones. This should not be confused with the probability of finding the global optimum by random choice in the LON which is $1/|V|$.

Now, comparing the results for a few different values of K, going from K low to high (see Fig. 4) we see that the number of optima increases at a fast rate with increasing K, a well known fact [17]. The frequency of visit to the global optimum (i.e., the last point on the right of the x-axis) is higher for low values of K where there are fewer optima and the walk often falls into the node having the best fitness or a close one. For instance, the image for $K = 2$ in Fig. 4 shows that the global optimum gets more than 70% of the occupation probability at equilibrium. For $K = 6$ the probability of being at the global optimum is already down to 0.10. We know (see [4] and Fig. 5) that the size of the basins shrinks with increasing K while their number increases quickly and, although the high-fitness basins

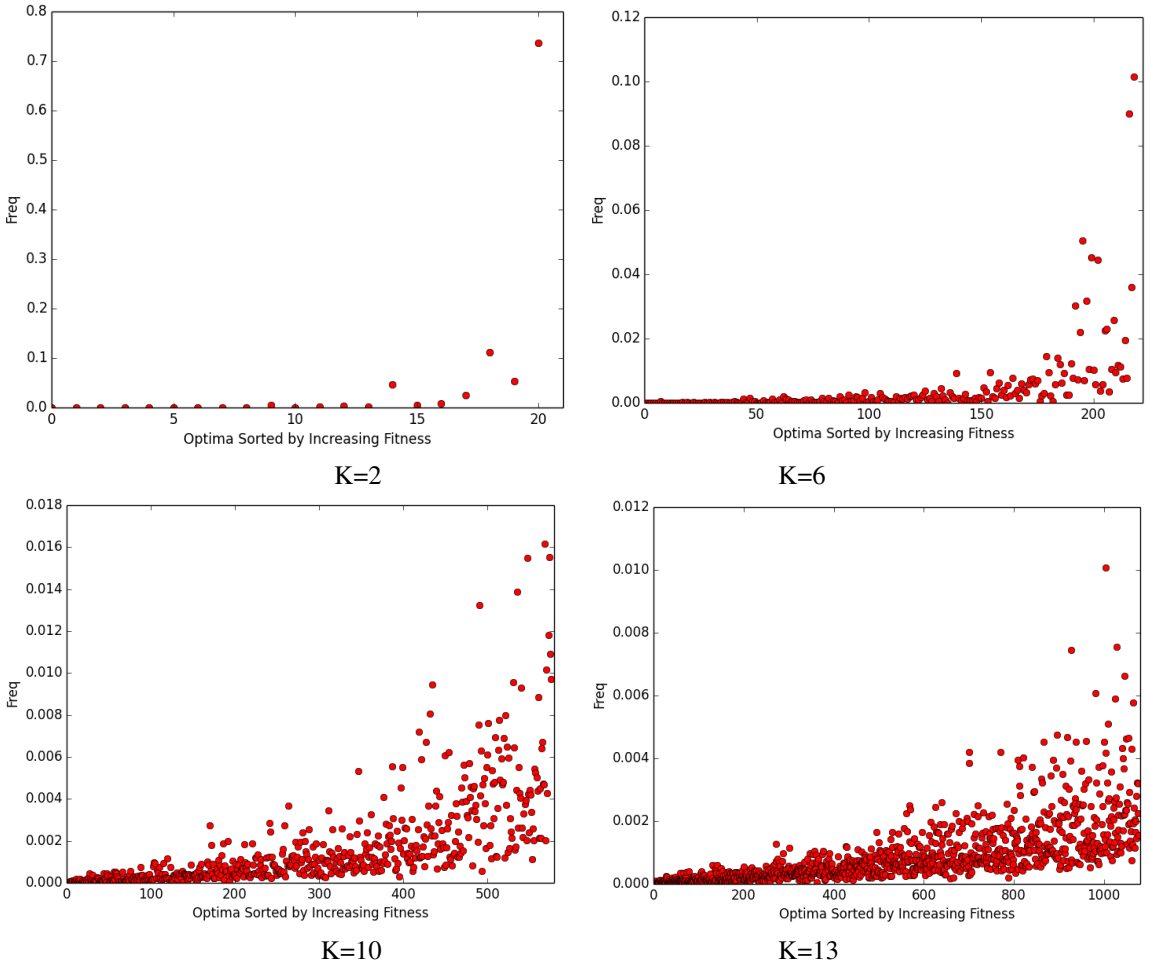

Fig. 4. Frequency of optima at equilibrium in the random walk for four instances with $N = 14$ and increasing K. Optima are ordered by increasing fitness.

are larger than the low fitness ones and the corresponding vertices in the LON have thus high in-degree, many high-fitness local maxima are frequently visited besides the best one, as we see in the lower images of Fig. 4.

This is confirmed by Fig. 6 which reports the frequency of the runs in which the most visited node is the global optimum averaged over 20 independent instances. We see that the frequency is higher for low K values but it decreases quickly for higher K and becomes zero or close to zero for the highest Ks. This is due to the combined effects of the rapid increase of the number of optima and of the stochasticity of the random walk process. For low K there are few optima and the basins of the best one is large, i.e., the corresponding LON node has many incoming edges and is a big attractor. For high K many good optima have large basins and this translates into many LON nodes having high in-degree. This shows that, since the random walk does not take the fitness of the nodes into account, random factors may decide which node is visited most. Nevertheless, it remains true in general that the best optima are visited more frequently than less good ones.

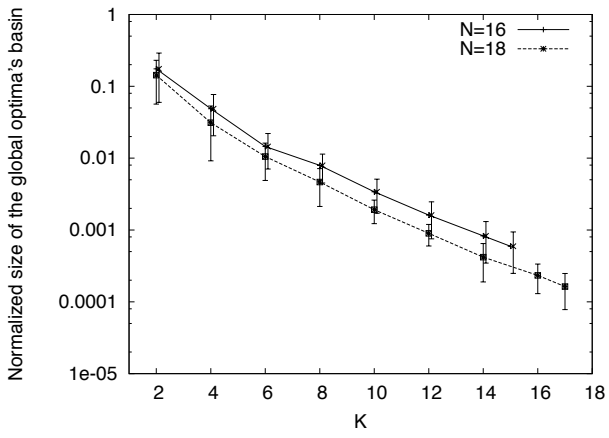

Fig. 5. Average over 30 independent landscapes of the normalized size of the global optimum basin for $N = 16$ and $N = 18$ for increasing values of K.

Fig. 7 illustrates another side of the coin. For the problem instances shown in Fig. 2 for $K = 12$ and in Fig. 4 with

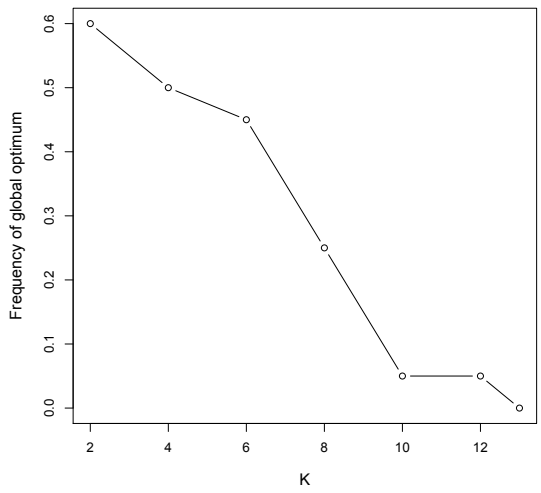

Fig. 6. Empirical probability for the global optimum to have the higher equilibrium occupancy in the random walk as a function of K for $N = 14$. The results are averages over 20 independent instances for each K.

$K = 6$, it gives the average first hitting time (see Sect. III), in time steps, for each optimum in the corresponding LON with the fitness of optima increasing towards the right of the x-axis. Intuitively one would expect that the best optima, which get many more visits as we saw above, should also be visited earlier in the average. This is confirmed by the results shown in Fig. 7 which have a sort of "reciprocal" shape with respect to the corresponding frequency plots. Indeed, an argument based on a mean-field approximation leads to a coarse first hitting time $t_{ij} \approx 1/p_j^*$ for vertex j starting from vertex i, where p_j^* is the equilibrium random walk frequency of vertex j [16]. Although the quantity plotted in Fig. 7 is, for a given target optimum j, the empirical average $\langle t_{ij} \rangle$ over all the starting nodes $i \neq j$, the inverse behavior is clearly visible.

From Fig. 7 we see that high-fitness optima are hit very quickly while low-fitness ones are in general hit for the first time after many random walk steps. Clearly, owing to the stochasticity of the process, some low-fitness local optima might get hit for the first time rather quickly but most will take a large number of time steps before being reached by the walk. This information is typically something that could not be inferred by running the PageRank algorithm, which is extremely efficient but does not directly provide any other information beyond the ranking of nodes.

B. Random walks on QAP landscapes

By definition, all the optima in an NK landscape have different real values, except for possible numerical roundoff errors; therefore they are strict maxima. On the other hand, combinatorial problems based on a permutation representation such as QAP and many others, usually contain variable, non-negligible amounts of neutrality, that is, regions of the fitness landscapes in which neighboring solutions have the same fitness. Such "plateaus" or "neutral networks" can sometimes be locally or globally optimal [7]. This is often also the case of QAP instances. Random walkers would wander for long times in such neutral networks. This would not be a problem in principle but, to avoid such complexities in a first study, the QAP instances examined here only have strict local minima. For both real-like and uniform instances we use a size of eleven, which still allows to compute LONs exhaustively without having to resort to sampling.

Fig. 8 shows the limiting frequencies of the random walks on a real-like instance (left image) and an uniform one (right image) arbitrarily chosen among the many that were examined. In general, they all show the same general patterns. We recall that, for the same problem dimension, here 11, uniform instances produce larger LONS, with many more optima, as can be observed on the x-axes of the figure, and that uniform random instances are more difficult to solve to global optimality than real-like ones of the same size. (see [21], [22]). As in NK landscapes and in other combinatorial optimization problems [7], the number of local optima increases quickly with problem dimension and the size of the corresponding basins decreases [21].

One sees that, in both cases, a minority of high-fitness optima are visited much more often that the rest, especially in the random uniform case. The trend is the same as what was observed for NK landscapes but it is more biased towards good optima. In fact, and differently from the NK case, the most visited optimum has always been the global one for the uniform instances, followed by the second best, for all the fifteen instances studied. For real-like instances this is often, but not always the case and, in fact, we observed some inversions. Due to the relatively low number of instances tried, we cannot statistically quantify the phenomenon in a significant way and so we only give the qualitative trend.

The mean first hit times follow the same approximate inverse relation with respect to visit frequency (see Fig. 9). However, in the QAP case the time distribution is more spread out and less regular than the NK case which, after all, are contrived random and isotropic functions.

As in NK landscapes and in many other difficult combinatorial optimization problems, the number of local optima increases quickly with problem dimension and the size of the corresponding basins decreases [7], [21] but, within this trend, the size of the basins is larger for better optima. In LONs, as we said in the previous section, this translates into more incoming links to better optima. The following Fig. 10 which is a scatterplot of incoming strength, i.e. the total weight of the incoming arcs to a node, against the frequency of that node in the random walk, clearly shows that there is a strong positive correlation. The figure refers to the instance whose frequency plot is given in the left image of Fig. 8. Logs are used because frequencies span six orders of magnitude. Thus, optima to which there is a high total transition probability are visited quickly and very often. This also means that, since self-loops have high weights [4], [21], once the walk reaches a strong optimum, it is likely to stay there for a while. For a local search algorithm this means that those same optima would be strong attractors.

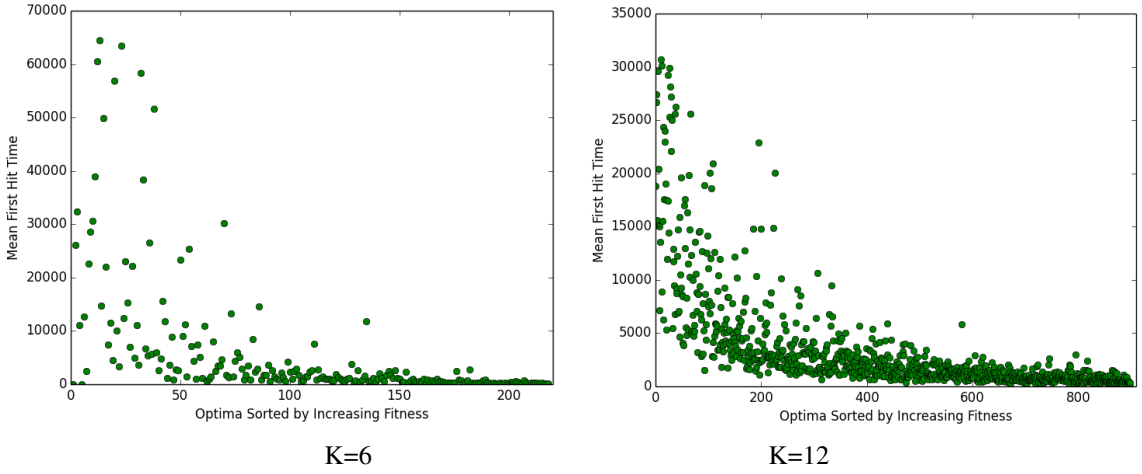

Fig. 7. Distribution of the mean first node hitting times with vertices of the LON ordered according to increasing fitness for two NK instance with $N = 14$ and $K = 6, 12$.

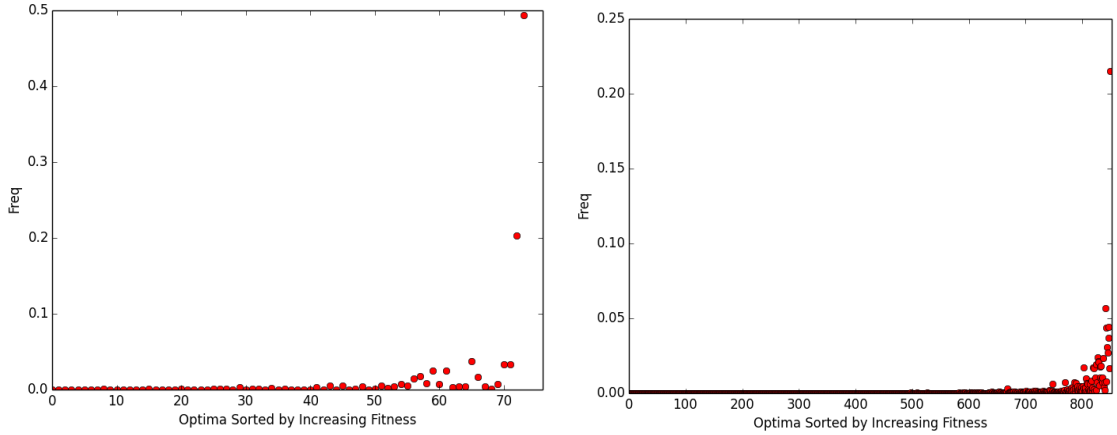

Fig. 8. Frequency of optima at equilibrium in the random walk for a QAP instance of the type real-like (left image) and an instance drawn from the set of uniform instances (right image). Optima are ordered by increasing fitness.

VI. Conclusions

Random walks have proved to be very useful to investigate the behavior of stochastic processes in many fields [16]. Here we applied them to the study of the local optima networks, or LONs, extracted by some typical combinatorial optimization problems. As case studies we used the NK landscapes and QAP instances of two different kinds. NK landscapes, though artificial, are useful because their difficulty can be controlled by the number of random site interaction K for a given N. On the other hand, QAP instances are typical of classical combinatorial optimization problems. In both cases we have shown that, in the long run, the best optima are visited more often thanks to the fact that their basins of attraction are larger. In the LON model this translates into nodes with a high number of incoming links which, in turn, is the reason why the walks reach them more easily. However, due to the exponential-like increase of the number of optima for growing K, or going from real-like to uniform random instances for QAP, the random walk process needs quickly increasing times to converge. Another aspect of the walks is the mean first hitting time, which can be computed numerically from our Monte Carlo simulations. These times should be theoretically in an approximate inverse relationship with the equilibrium frequency of visit of each local optimum. Indeed, that the relationship holds empirically in all cases, although the randomness of the numerical process does not allow to draw more precise inferences. Finally, random walks are useful but they are blind at the local optima level and do not resemble local search techniques in common use. In the future, it would be interesting to study the behavior of adaptive walks, including diffusion in the case of neutrality, which take fitness into account. Some simple local search techniques such as iterated local search and simulated annealing on LONs would also be interesting to extract information about typical fitness barriers or perturbation strengths to be used in those metaheuristics.

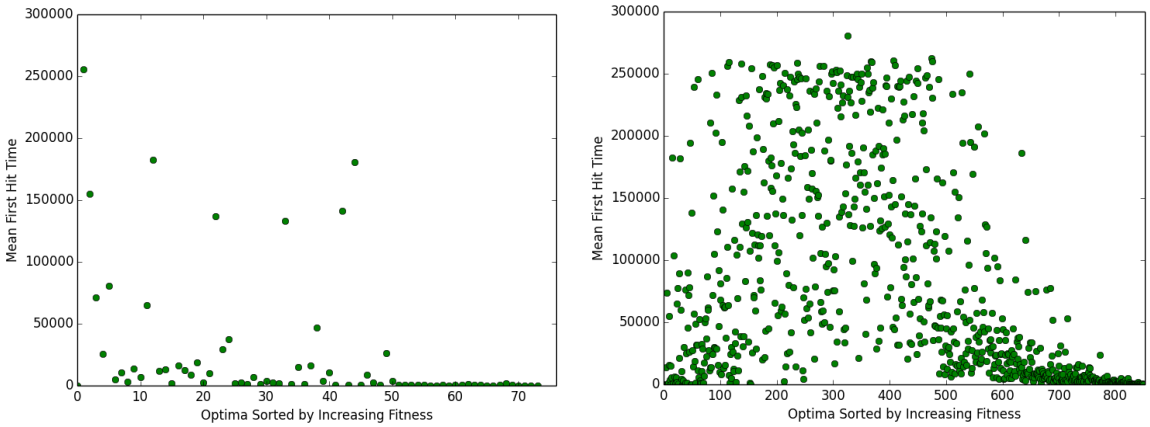

Fig. 9. Distribution of the mean first node hitting times with vertices of the LON ordered according to increasing fitness for the two QAP instances of Fig. 8.

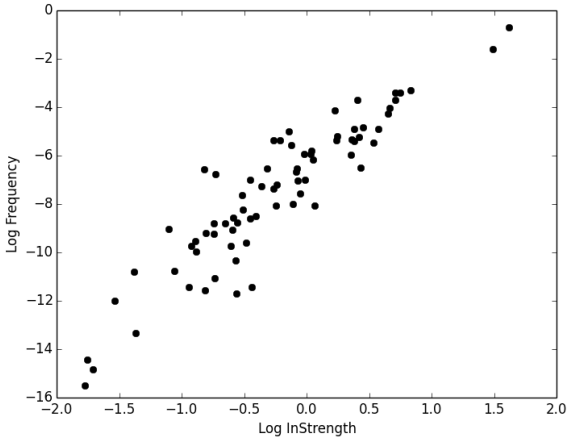

Fig. 10. Scatterplot of the incoming strength into a node against the limiting visit frequency of that node, with nodes ordered according to increasing fitness. The data refer to the instance whose frequency plot is shown in the left image of Fig. 8.

ACKNOWLEDGMENT

I am indebted to S. Verel, G. Ochoa, and F. Daolio for providing the program code for computing local optima networks and for many useful discussions during our collaboration.

REFERENCES

[1] E.-G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.

[2] H. Richter and A. E. (Eds.), *Recent Advances in the Theory and Application of Fitness Landscapes*, ser. Series in Emergency, Complexity, and Computation. Berlin: Springer, 2014.

[3] G. Ochoa, M. Tomassini, S. Verel, and C. Darabos, "A study of NK landscapes' basins and local optima networks," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2008*. ACM, 2008, pp. 555–562.

[4] M. Tomassini, S. Verel, and G. Ochoa, "Complex-network analysis of combinatorial spaces: The NK landscape case," *Phys. Rev. E*, vol. 78, no. 6, p. 066114, 2008.

[5] S. Verel, F. Daolio, G. Ochoa, and M. Tomassini, "Sampling local optima networks of large combinatorial search spaces: the QAP case," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2018, pp. 257–268.

[6] S. Verel, G. Ochoa, and M. Tomassini, "Local optima networks of NK landscapes with neutrality," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 6, pp. 783–797, 2011.

[7] L. Hernando, A. Mendiburu, and J. A. Lozano, "Anatomy of the attraction basins: Breaking with the intuition," *Evolutionary Computation*, pp. 1–32, 2018.

[8] M. E. J. Newman, *Networks: An Introduction*. Oxford University Press, Oxford, UK, 2018.

[9] S. Herrmann, G. Ochoa, and F. Rothlauf, "Pagerank centrality for performance prediction: the impact of the local optima network model," *Journal of Heuristics*, vol. 24, no. 3, pp. 243–264, 2018.

[10] S. K. Thompson and O. Frank, "Model-based estimation with link-tracing sampling designs," *Survey Methodology*, vol. 26, no. 1, pp. 87–98, 2000.

[11] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[12] P. Pons and M. Latapy, "Computing communities in large networks using random walks." *J. Graph Algorithms Appl.*, vol. 10, no. 2, pp. 191–218, 2006.

[13] E. Weinberger, "Correlated and uncorrelated fitness landscapes and how to tell the difference," *Biological cybernetics*, vol. 63, no. 5, pp. 325–336, 1990.

[14] C. Reidys and P. Stadler, "Combinatorial landscapes," *SIAM review*, vol. 44, no. 1, pp. 3–54, 2002.

[15] S. Verel, F. Daolio, G. Ochoa, and M. Tomassini, "Local optima networks with escape edges," in *Proceedings of the International Conference on Artificial Evolution, EA-2011*, ser. Lecture Notes in Computer Science, vol. 7401. Springer, 2012, pp. 49–60.

[16] N. Masuda, M. A. Porter, and R. Lambiotte, "Random walks and diffusion in networks," *Physics Reports*, vol. 716, pp. 1–58, 2017.

[17] S. A. Kauffman, *The Origins of Order*. New York: Oxford University Press, 1993.

[18] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica*, vol. 25, no. 1, pp. 53–76, 1957.

[19] J. Knowles and D. Corne, "Instance generators and test suites for the multiobjective quadratic assignment problem," in *Proceedings of the Evolutionary Multi-Criterion Optimization Conference (EMO 2003)*, ser. LNCS, no. 2632. Springer, 2003, pp. 295–310.

[20] E. D. Taillard, "Comparison of iterative searches for the quadratic assignment problem," *Location Science*, vol. 3, no. 2, pp. 87 – 105, 1995.

[21] F. Daolio, S. Verel, G. Ochoa, and M. Tomassini, "Local Optima Networks of the Quadratic Assignment Problem," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[22] G. Ochoa and S. Herrmann, "Perturbation strength and the global structure of QAP fitness landscapes," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2018, pp. 245–256.